VERITAS

Veritas Alta Application Resiliency

Azure integration and implementation.

White Paper | March 2023

VERITAS ALTA™

Contents

Introduction

Microsoft Azure is a leading public cloud platform that offers several different options for using IaaS resources as the basis for running applications of all types. This includes those of high importance that have strict RPO/RTO requirements. Veritas Alta™ Application Resiliency provides the high availability and disaster recovery solution required for applications with high uptime requirements that can't be met using native Azure tools alone. Using these best practices and configuration guidance, Veritas Alta Application Resiliency will manage the Azure infrastructure components to ensure that the application stack deployed on Azure infrastructure is highly available and can be managed to avoid application downtime in the event of a service disruption or outage.

There are several key benefits realized when using Veritas Alta Application Resiliency to manage application high availability in Azure environments:

- Instant fault detection – The Azure-specific Veritas agents ensure that application downtime is minimal in the event of a system-level disruption or failure
- Geographic redundancy – with the Veritas' Global Cluster Option (GCO), applications can be configured for high availability across Azure zones and regions, minimizing the likelihood of application downtime in the event of a local or regional outage
- Replication – Veritas Volume Replicator (VVR) manages the data movement between Azure zones and regions with a zero data loss architecture and flexible configuration options that minimize Azure resource utilization
- Management simplicity – VIOM provides a single interface for visibility and control of the Veritas managed resources within the Azure environment, as well as reporting on events and potential risks to application uptime
- Build shared storage clusters using Azure native compute and storage infrastructure that significantly increases application performance and data resiliency

With flexible configuration options that are designed to integrate with native Azure components, Veritas Alta Application Resiliency enables users to deploy highly available applications in Azure environments with the confidence that they are being managed to provide maximum uptime for your business services.

Scope

This document intends to guide the reader in implementing Veritas Alta Application Resiliency on Linux on an Azure platform and assumes some pre-existing knowledge of the Veritas high availability and disaster recovery solution. Guidance is provided on the Veritas Azure-specific agent and, replication set up in Azure.

First, an overview of the solution as configured in Azure is discussed, including the components required and the order of configuration.

Next, each relevant component of the solution is reviewed, and the configuration is described, including the following:

1. Storage
2. Networking
3. Veritas Cluster Server

Understanding the configurations of Global Cluster Option, Veritas Volume Replication, and Veritas Cluster Server enable a better understanding of this paper. It is not intended as an introduction to Veritas Alta Application Resiliency for new users. For more information on these features, refer to the official documentation found here:

- Veritas InfoScale 8.0 Disaster Recovery Implementation Guide - Linux
- Veritas InfoScale 8.0 Solutions in Cloud Environments
- Veritas InfoScale 8.0 Replication Administrator's Guide - Linux

Solution Overview

The solution in Figure 1 shows two Veritas clusters making use of Veritas Volume Replicator to replicate data from one cluster to another. The Global Cluster Option is configured to enable failover from one cluster to the other. Each cluster is located in a separate Azure Virtual Network. Virtual Networks (vnet) are peered so that traffic can route between them.

Storage devices cannot be shared in Azure. Once failover is detected, Veritas Cluster Server (VCS) orchestrates a failover operation where the disks are detached from the failed node and reattached to the failover node.

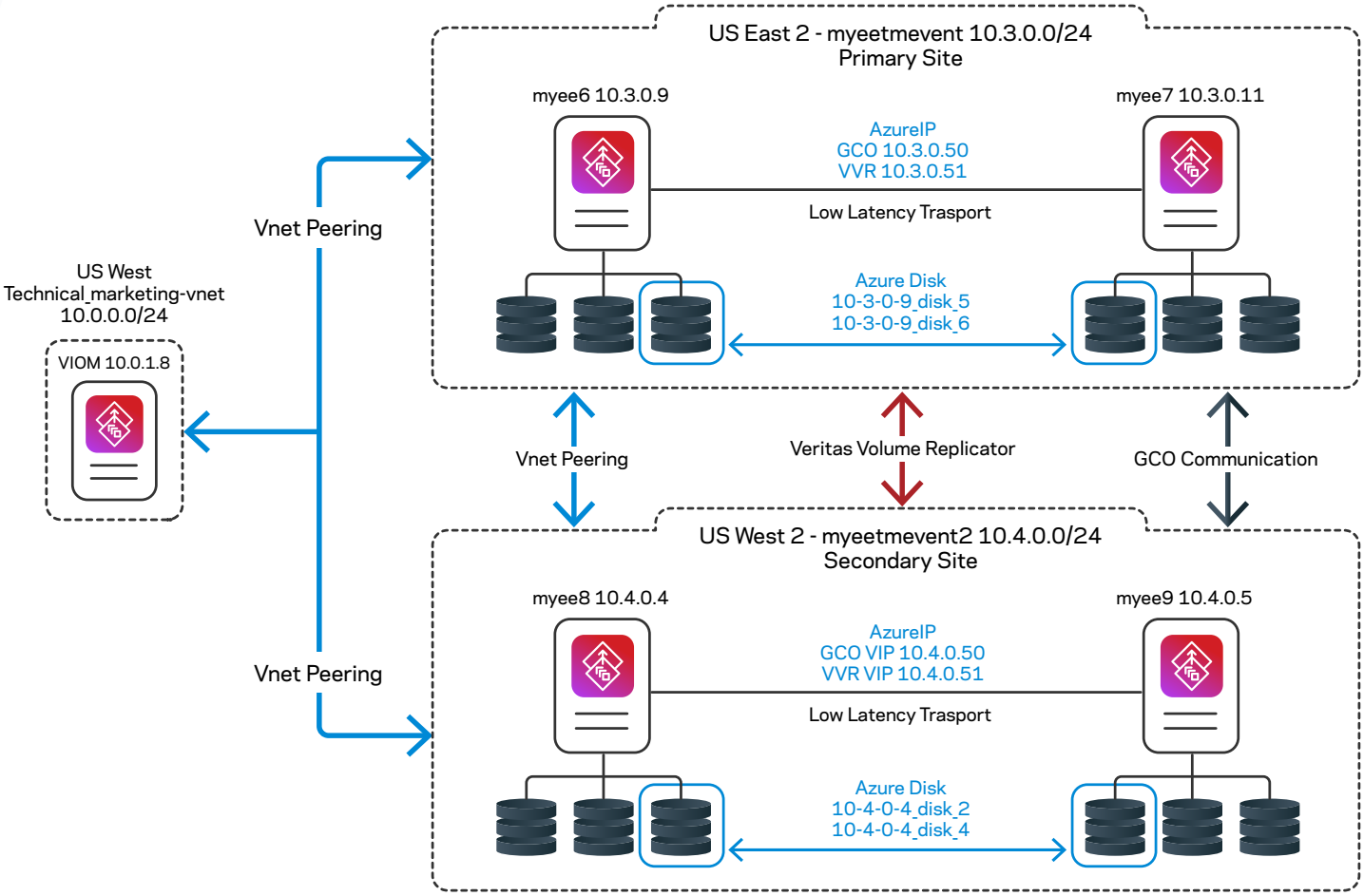


Figure 1 - Solution Diagram

Component Overview

The environment includes the following components and topology:

1. Two Veritas clusters, each is running on RHEL.
2. Each cluster is running in a separate geographical region, connected by vnet peering.
3. One cluster is replicating to the other with Veritas Volume Replication (VVR).
4. Each cluster is using Azure managed disks.
5. Both clusters are managed by Veritas InfoScale Operations Manager (VIOM) in a third geographical region, also connected by vnet peering.
6. Global Cluster Option (GCO) enables the failover between the cluster in the primary site to the cluster in the secondary site (In Figure 1 GCO failover is between the cluster in US East 2 and the cluster in US West 2).

Some notes about the environment:

- VIOM is not mandatory for this solution, nor is its location in a third geographical region. This particular configuration is intended as an example of a possible deployment scenario where the Veritas Alta Application Resiliency components are not co-located within the same Azure regions
- This solution focuses on a simple clustered file server with replication. Applications can be configured to run on this system and can be configured with the same resiliency and disaster recovery features
- Flexible Storage Sharing (FSS) and Veritas Cluster File System (CFS) can be used to create a scalable, highly available, performant file system that may substitute the AzureDisk storage subsystem presented in this environment
- Within a region, Azure permits the configuration of VMs in specific availability zones (AZ). For additional high availability, each node in a cluster may be configured to use a different AZ. AZ configuration is abstracted from the networking topology, so no additional Veritas Alta Application Resiliency configuration is needed
- Similar to Availability Zones, Azure Availability Sets help increase the availability and reliability of VMs with Availability sets. VNETs can span Availability Sets, so no additional networking configuration needs to be implemented within Veritas Alta Application Resiliency

Configuration Order

It is recommended that the configuration of the solution is done in a particular order to ensure that tasks may proceed without unnecessary interruption. Configuration in this order is not a requirement, but it will be less time-consuming.

1. Set up Virtual Networks (vnets) if more than one is required
2. Configure vnet peering so that traffic can route between the two
3. Create managed disks
4. Create VMs
5. Install Azure Python SDK (see Cluster Server 8.0 Bundled Agents Reference Guide - Linux) on all participating nodes
6. Install InfoScale without the Global Cluster Option (GCO)
7. Configure static IP addresses for the VMs
8. Enable GCO
9. Bind AzureIP agent (in VCS) to the desired GCO IP Address
10. Configure the AzureDisk agent (in VCS) to bind the particular managed disks to the desired VMs
11. Configure filesystem (including all the required volumes)
12. Configure cluster-to-cluster replication with VVR
13. Configure VCS to manage VVR
14. Configure VCS to manage cluster-to-cluster replication

Storage

Initial storage configuration with Azure managed disks requires the disks to be connected to a single node, configured with Veritas Volume Manager in a disk group and formatted with a Veritas filesystem (vxfs). Then, the managed disk(s) are configured within VCS with the AzureDisk agent to enable high availability so that the disk(s) can be 'moved' automatically to a failover node and back again. Moving a disk from a primary to a failover node(s) essentially means that the disk is detached from the primary node and re-attached to the failover node. This process is managed by the VCS AzureDisk agent.

Azure storage is managed via RESTful API calls and can be created and manipulated via (but not exclusively by) the Azure UI or Azure CLI.

Azure Managed Disk

Azure managed disks are created as objects outside of the task of creating a virtual machine. They are also portable between virtual machines (with certain restrictions). Azure managed disks can be connected and disconnected from VMs using the Azure API. Depending on the functionality required, Azure un-managed disks may also be used. For more information on the difference between the two types of disks, refer to the BUILDWINDOWS blog.

Azure managed disks come in a variety of performance tiers. Specific tiers of performance are only available in certain geographical regions.

Veritas AzureDisk Agent

The Veritas AzureDisk agent is used by VCS to connect and disconnect Azure Managed Disks to VMs. During failover within a Veritas cluster, an AzureDisk or group of AzureDisks is detached from the failed VM and re-attached to another VM in the cluster.

The AzureDisk Agent configuration in VCS depends on other VCS resources, and the configuration is discussed in the Veritas Cluster Server section later in this document.

Note that Flexible Storage Sharing (FSS) can also be configured so that FastFailover is available. FSS combined with Cluster File System (CFS) enhances the cluster by reducing the amount of time required to perform a failover. For more information on FSS, see Storage Foundation Cluster File System High Availability 8.0 Configuration and Upgrade Guide - Linux.

Networking

Veritas Alta Application Resiliency's networking topology in Azure is almost the same as a regular network with the exception of an additional requirement where IPs, networks, and NICs are treated as programmatic objects. This enables greater flexibility in managing a network in an Azure environment. Veritas Alta Application Resiliency supports Azure networking with the VCS Azure IP agent so that it can integrate seamlessly and transparently to provide high availability to geographically (and locally) clustered applications.

Virtual Network (VNET)

One of Microsoft Azure's necessary network objects is known as the Virtual Network or Vnet. The vnet contains an address space and subnets. A vnet is contained within a geographical region. VMs residing in a vnet can communicate with VMs in other vnets. Vnets must be connected with vnet peering, demonstrating that Azure is capable of automatically correcting a circular network configuration.

Vnet Peering

In this example configuration, vnet peering is configured between 2 vnets in different geographical regions. The VIOM server resides in a third region, also connected to the two other vnets through vnet peering.

Network Interface Configuration

During the initial configuration of a new VM, it is only possible to configure one network interface. The Veritas best practice requires provisioning of 3 network interfaces as described in Table 1. Be aware that specific sizes of Virtual Machines determine the number of NICs that can be connected to them.

Table 1 Veritas Alta Application Resiliency Networking Interfaces

Network Interface	Purpose
1	Public Network Traffic
2	Low Latency Transport Link
3	Low Latency Transport Link

To add additional Network Interfaces, the VM must be shut down and deallocated. Additional network interfaces must be created, and subsequently attached to the VM, before it can be powered up again. In this solution, the Low Latency Transport (LLT) links were configured on subnets distinct from the public network. It is not possible to create and connect separate VNETs to a VM.

For more information on Veritas Alta Application Resiliency networking, please see Cluster Server 8.0 Administrator's Guide.

IP Addresses

Microsoft Azure IP addresses are objects that must be explicitly allocated to Azure infrastructure such as virtual machines (VM). An IP address is automatically allocated to an Azure VM's network interface (NIC). A typical sysadmin's task is to allocate a second IP address to a NIC. In Azure, this operation can be completed; however, the second IP address will be unusable. The correct method is to assign the second IP address to the NIC object through the Azure management interface (or Azure API).

IP addresses are not configured as static by default in Azure. When provisioning a VM, the IP address assigned must be configured as static after it has been provisioned. Ensure that all the interfaces attached to the VM are also statically assigned.

Figure 2 shows a high-level network topology of the solution depicted in Figure 1.

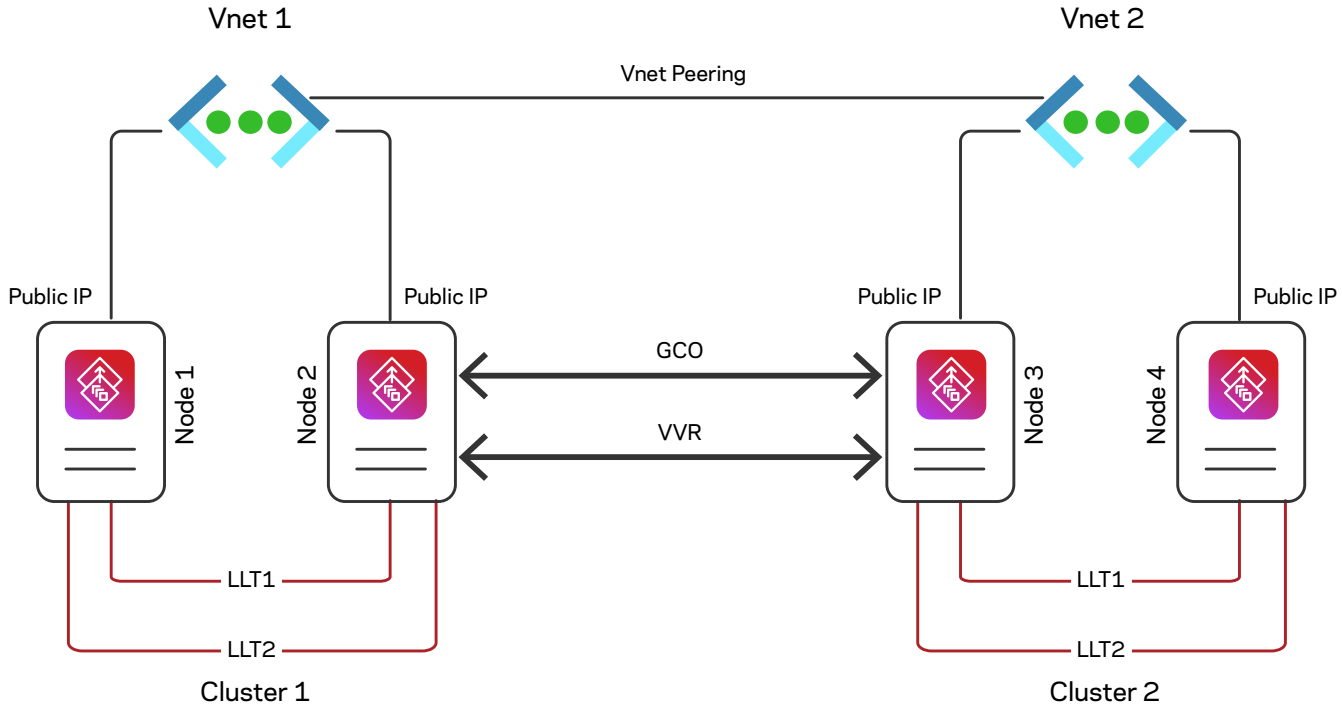


Figure 2 Networks and IP Addresses

Per Figure 2, the following is a list of IP addresses required per node.

1. Public network IP. This is required for public network traffic (such as connections from client workstations)

2. Low Latency Transport IP 1
3. Low Latency Transport IP 2

Each cluster requires the following IPs:

1. Global Cluster Option (GCO) IP
2. Veritas Volume Replication (VVR) IP

The GCO and VVR IPs must be configured as AzureIP resources in VCS (with the AzureIP agent), so they can be failed over between nodes in a single cluster.

Low Latency Transport Optimization

Low Latency Transport networking links can be tuned to optimize performance. In particular, configure LLT to make use of the UDP protocol, during installation. As well, after completing the InfoScale installation, modify the LLT configuration to make use of the following parameters in Table 2.

Table 2 Low Latency Transport Tuning

Parameter(s) Modify	Definition
In "/etc/llttab", add the following parameters: set-flow window:20 set-flow highwater:10000 set-flow lowwater:8000 set-flow rporthighwater:10000 set-flow rportlowwater:8000 set-flow ackval:2 set-flow linkburst:32 Ensure that the LLT connections are configured to make use of an MTU of 1500.	These parameters tune the network characteristics of the LLT links.
In "/etc/sysconfig/llt", add the following parameters: LLT_ENABLE_AWINDOW=0 LLT_NMULTIPORTS=2	The AWINDOW parameter enables the LLT adaptive window feature. NMULTIPORTS enables higher concurrency in LLT connections..

Global Cluster Option (GCO) IP AzureIP Configuration

The Global Cluster Option is required to configure High Availability between two separate Veritas clusters. Each cluster participating in the Global Cluster requires a virtual IP in the form of an AzureIP to enable communication between them.

Veritas Volume Replication (VVR) AzureIP Configuration

Veritas Volume Replication requires a virtual IP for replication. Like GCO, the VVR virtual IP is an AzureIP object and must be allocated in advance. To configure VVR, provision the VVR IP to a specific primary node and a second VVR IP to a secondary node. Then proceed to configure VVR using the standard documented configuration procedure for VVR (Veritas InfoScale 8.0 Replication Administrator's Guide - Linux).

Veritas Cluster Server

Figure 3 shows the VCS configuration for a cluster with Global Cluster Option Failover and the constituent VCS Service Groups and Agents. Agents specific to Azure include:

1. AzureAuth
2. AzureDisk
3. AzureIP

The ClusterService service group is created automatically when GCO is enabled.

The configuration order for these Service Groups is as follows:

1. Configure the AzureAuth Service group in VCS
2. Set up the second cluster and install InfoScale
3. Enable GCO on the primary and secondary clusters (successive steps in this procedure must be performed on both clusters)
4. Add an AzureIP resource to the ClusterService service group. This IP will be the virtual IP required for the GCO heartbeat (wide area communication or wac)
5. Create managed disks in Azure (including those required for Veritas Volume Replication).
6. Configure the Storage service group (for the purpose of this paper, this was a discrete service group. AzureDisk resources are likely required inside application service groups).
7. Configure Veritas Volume Replication (create Replication Volume Group) and synchronize the clusters.
8. Configure the Replication Volume Group (RVG) resource in VCS.
9. Configure the Global service group for managing the failover from one cluster to another.

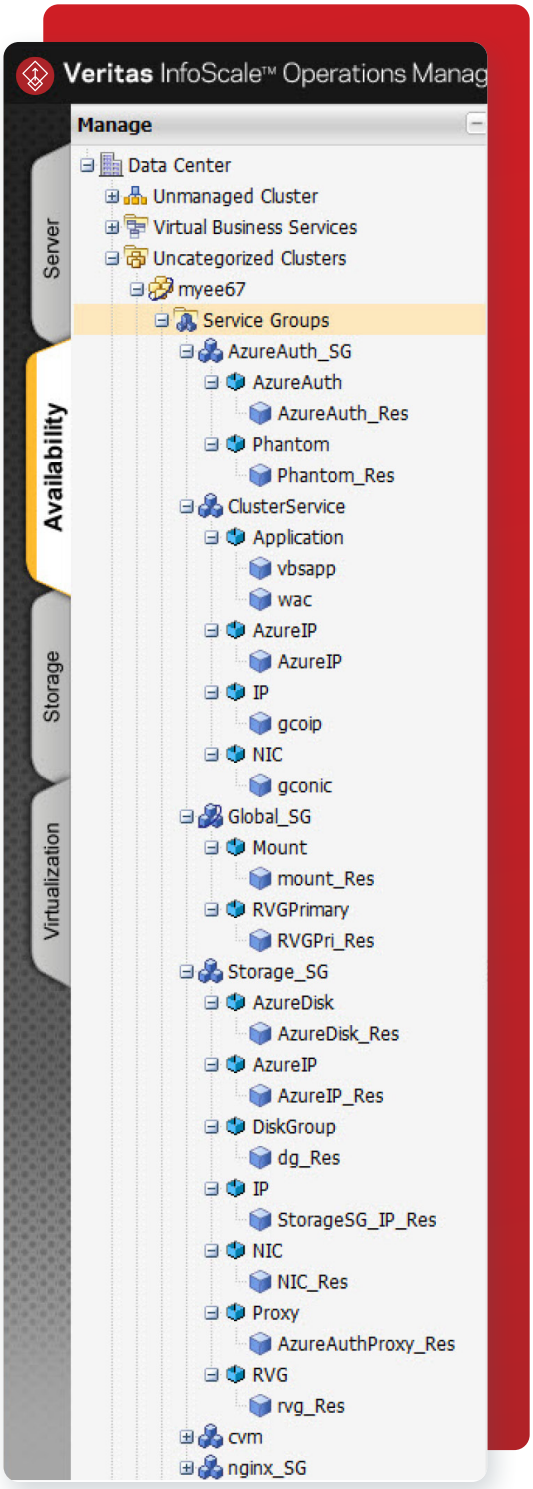


Figure 3 VCS Configuration

VCS Azure Agents

AzureAuth

To perform any operation on Azure resources, such as updating a resource record set, attaching an Azure data disk, assigning a private IP to a Network Interface, and any other operation requires authentication. The AzureAuth agent authenticates the Azure subscription using service principal credentials.

AzureAuth agent is a persistent resource that monitors the validity of service principal credentials.

To ensure the most uncomplicated experience when configuring Azure agents, test these credentials using SDK related tools like Azure CLI.

For more information on the AzureAuth agent, please see the Bundled Agents Reference Guide (Veritas Cluster Server 8.0 Bundled Agents Reference Guide - Linux) page 265.

AzureDisk

Virtual machines in Azure use data disks to store the application's data. The AzureDisk agent supports managed and unmanaged data disks and provides high availability of these disks during the fail-over of an application. The AzureDisk agent brings online, takes offline, and monitors the managed and unmanaged Azure data disks. It attaches the managed and unmanaged data disks to a virtual machine of the same or different resource group. The AzureDisk agent uses the Azure Python SDK to determine whether the Azure data disks are attached to the Azure virtual machines or not.

For more information on the AzureDisk agent, please see the Bundled Agents Reference Guide (Veritas Cluster Server 8.0 Bundled Agents Reference Guide - Linux) page 101.

AzureIP

The AzureIP agent manages networking resources in an Azure environment. There are multiple options that depend on the network topology required. In the solution in Figure 1, private IPs are used for VVR and GCO IPs on each cluster. For more information on the use of other types of AzureIP resources, please see the Bundled Agents Reference Guide (Veritas Cluster Server 8.0 Bundled Agents Reference Guide - Linux) page 172.

Like the AzureDisk agent, the AzureIP agent also uses the Azure Python SDK to perform operations with Azure.

Note that while the nomenclature in the VCS configuration for AzureIP refers to a Private IP, the context includes network traffic external to Azure. In the solution discussed in this paper and shown in Figure 1, the GCO and VVR IPs are within the VNET subnets, which are private to the Internet, but for the sake of discussion, they are in subnets where IP traffic can come from client machines and are thus 'public.'

Global Cluster Option (GCO) Configuration

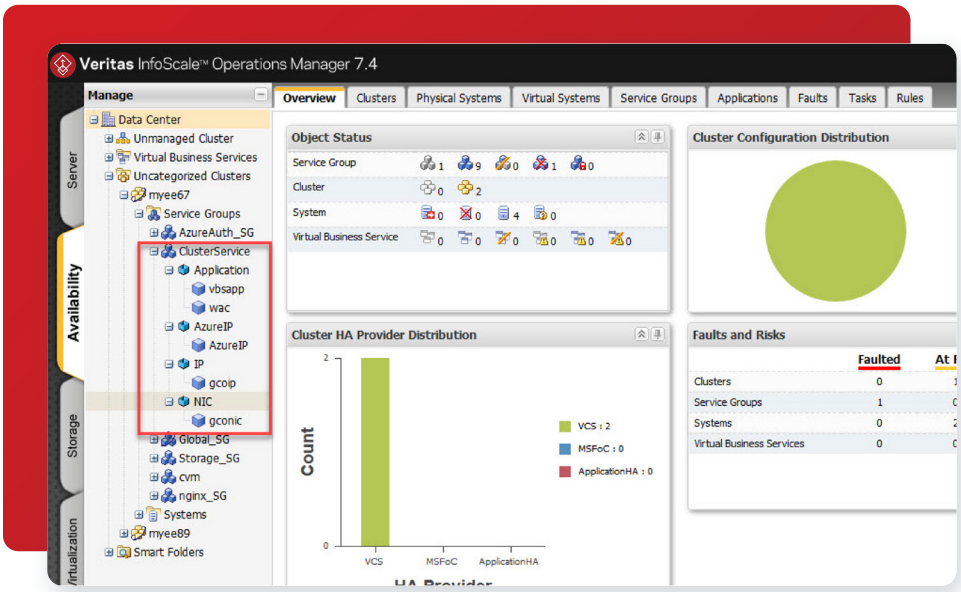


Figure 4 Example ClusterService Service Group

Veritas Global Cluster Option enables High Availability failover between different clusters. Local clustering provides local failover which would typically be for systems provisioned within the same cloud availability zones. Campus and replicated cluster configurations offer protection against disasters that affect limited geographic regions. In the cloud, this could be an outage of an availability zone or multiple availability zones. Large scale disasters such as major floods, hurricanes, and earthquakes can cause outages for an entire city or region – which could mean an entire cloud service provider region could be affected. In this situation, you can ensure data availability by migrating applications to sites located in different cloud regions.

For more information on configuring Global Clusters, please see the Cluster Server 8.0 Administrator's Guide

GCO Network Configuration

GCO configuration in VCS is contained within the built-in ClusterService service group. To configure GCO in Azure, the wide area connector service communicates over a virtual IP address that can be assigned to any node in the cluster.

The setup process for GCO (and VVR, which also requires a virtual IP address) requires that the virtual IP is assigned to a public NIC on a selected node for each of the clusters. This enables the virtual IPs, and from here, the GCO configuration can proceed. For example, Figure 5 shows IPs are configured on myee6 and myee8. GCO (and VVR) can be configured between those nodes. Then the corresponding AzureIP resources can be configured in their respective service groups.

An AzureIP resource must be configured in the ClusterService service group (For example, see Table 3 ClusterService).

Veritas Volume Replication Configuration

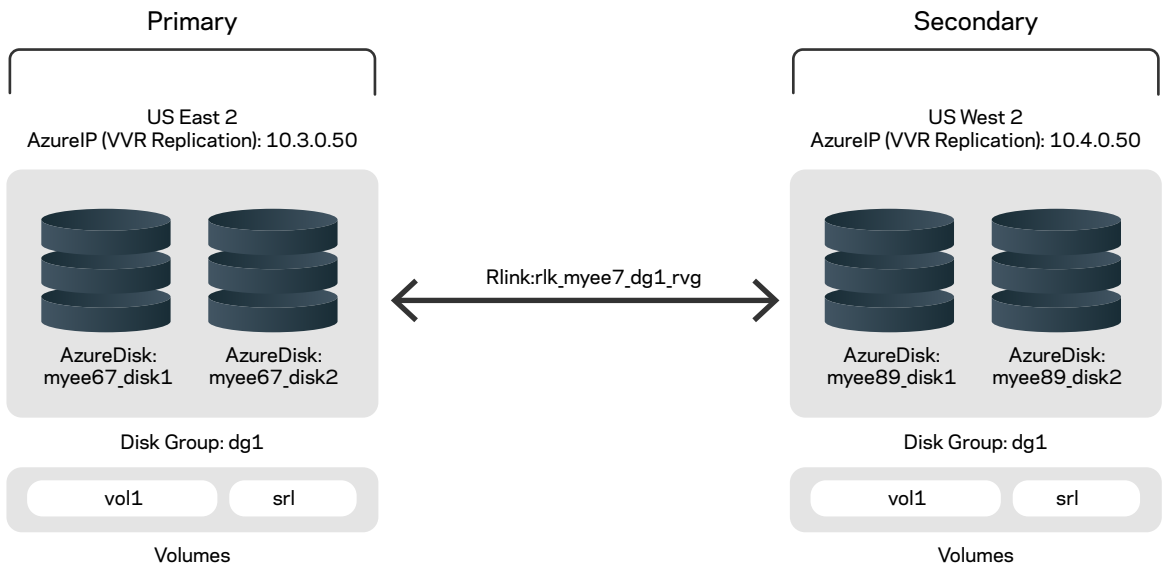


Figure 5 Veritas Volume Replication Configuration

Figure 6 shows how the disks are configured and the volumes used in this paper's environment for Veritas Volume Replication (VVR). The Replication Volume Group contains vol1 and the SRL (storage replication log). For more information on configuring VVR, see Veritas InfoScale 8.0 Replication Administrator's Guide - Linux.

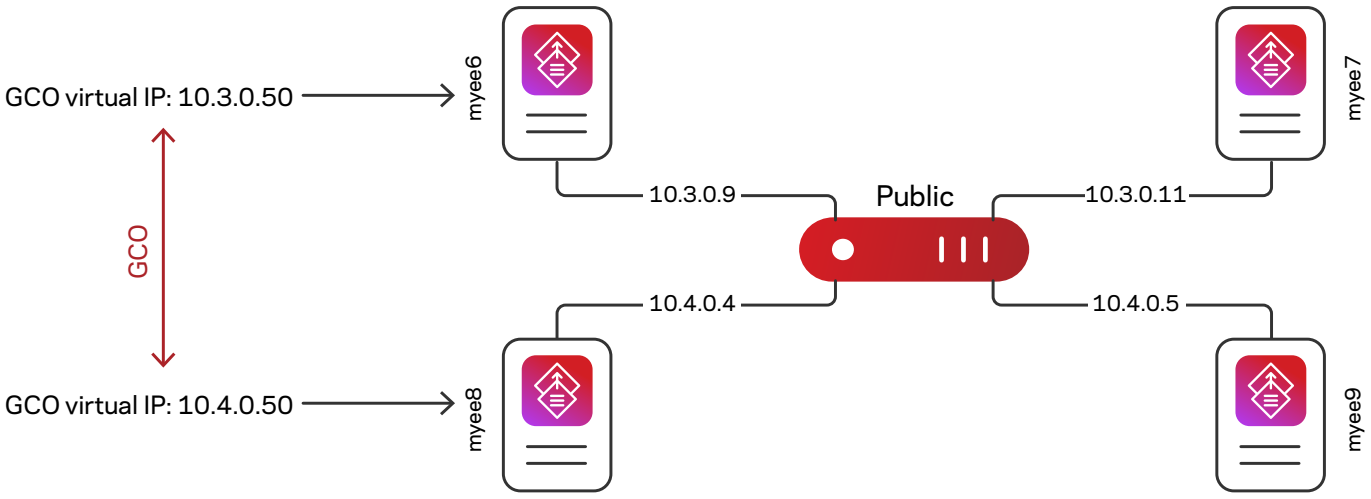


Figure 6 GCO initial network configuration

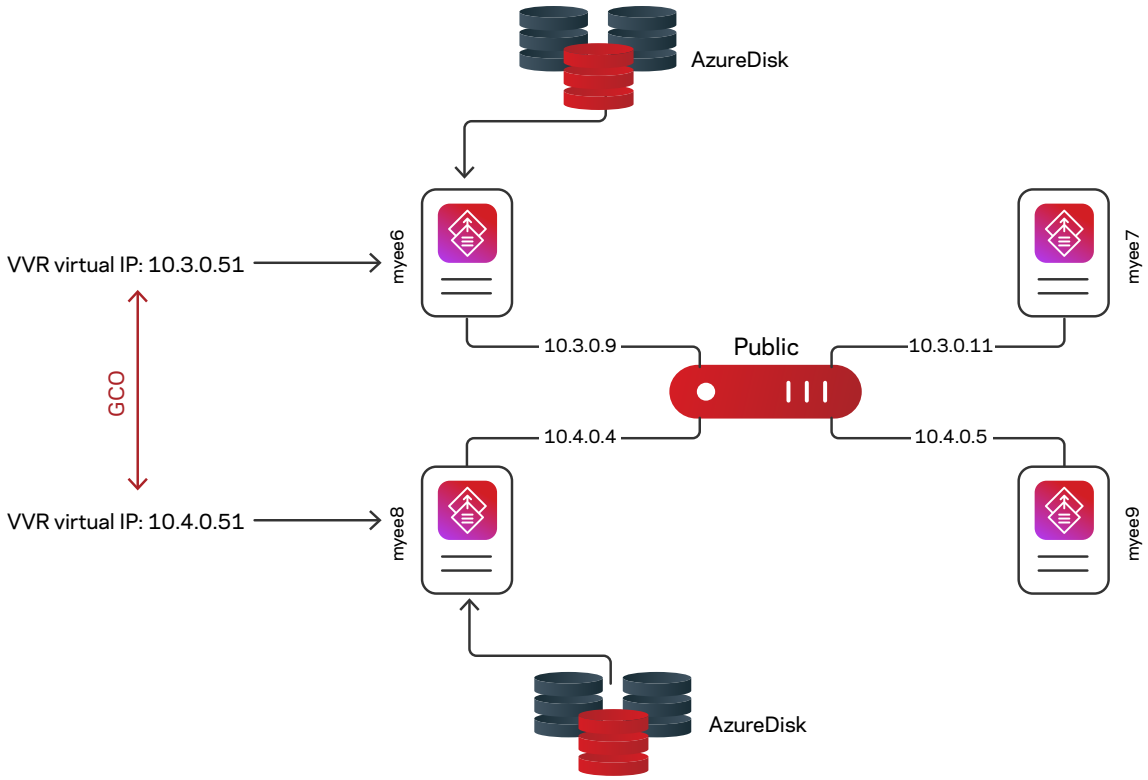


Figure 7 VVR initial configuration with AzureIP and AzureDisk

The storage and the virtual IP are Azure objects and managed by the VCS Azure agents (AzureDisk and AzureIP). Storage is failed over between nodes in the cluster, by VCS (for example, between myee6 and myee7, as shown in Figure 7) by detaching and attaching to a failover node. The VVR virtual IP is failed between nodes in a cluster by deregistering it from the failed VM, unconfiguring the IP from the NIC on the failed VM (if possible), and then registering and configuring the virtual IP on the failover node. Failover between clusters makes use of VVR to replicate data between sites.

To configure VVR, the managed disks and IP objects must be created in Azure and connected to specified nodes. In Figure 7, the AzureDisks and IP are connected to myee6 and myee8, respectively. After these objects are connected, the VVR configuration can be performed using the standard documented configuration procedure for VVR (Veritas InfoScale 8.0 Replication Administrator's Guide - Linux).

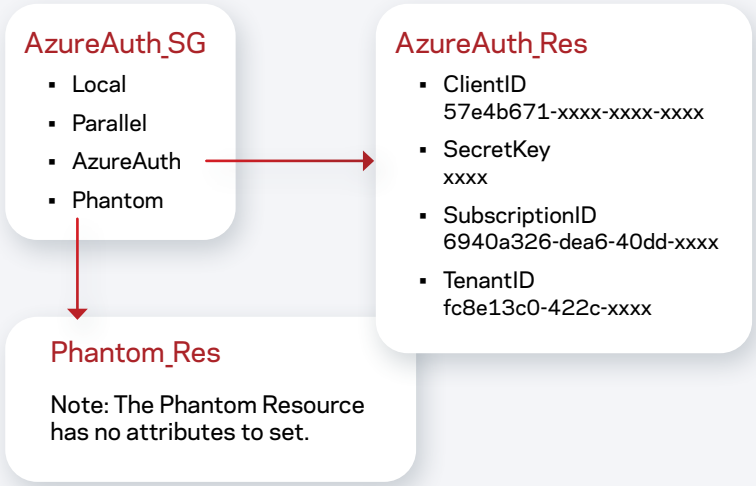
After VVR has been configured and a Replication Volume Group (RVG) created, a RVG resource must be configured in VCS in the required service group. The service group in which the RVG resource is contained is determined by the application, which must be protected with high availability. In this solution, there is no particular application other than a storage service. Please see Table 3 Storage Service Group for more information.

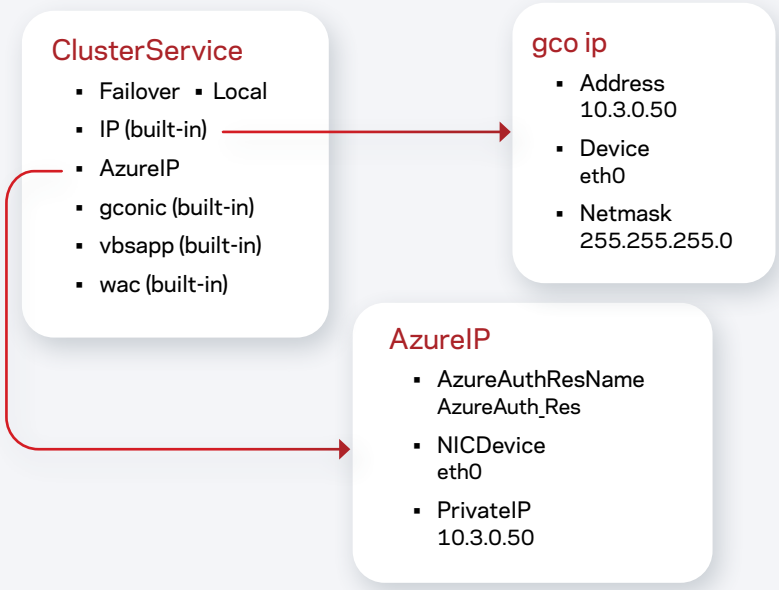
To configure Global Cluster Failover, a Global Service Group must be created, containing the RVG resource and, in this particular case, a resource for the storage mount. Please see Table 3 Global Service group for more information.

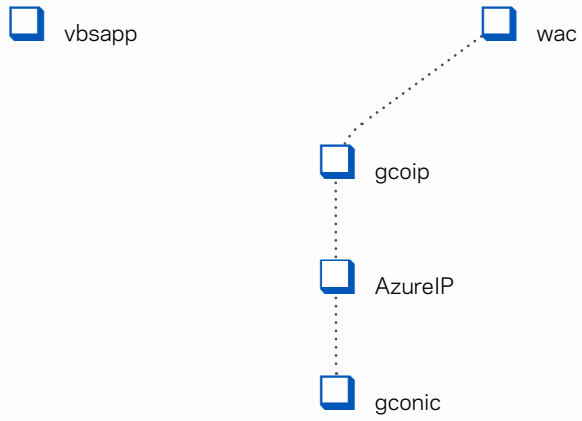
VCS Configuration and Dependencies

The following is a description of the configuration of the service groups required to make this solution work.

Table 3 VCS Configuration and Dependencies

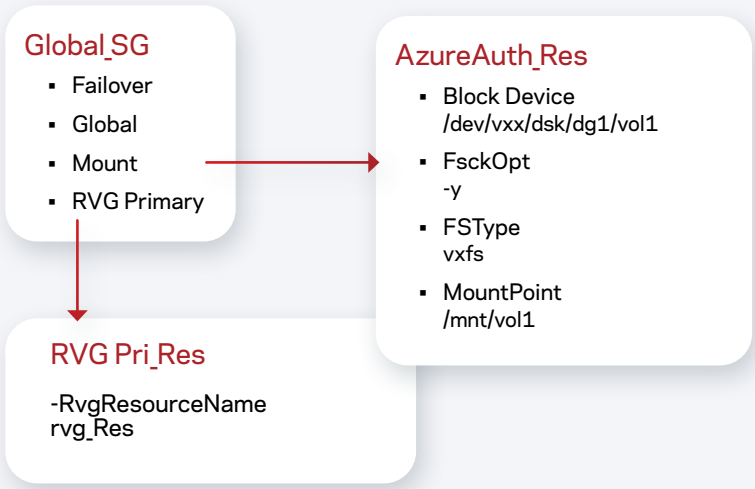
<table>
<tr><th colspan="2">AzureAuth</th></tr>
<tr><td> 
</td><td>The AzureAuth service group contains the AzureAuth resource, which is used to configure Azure account information so that RESTful API calls by other Azure Agents can be authenticated.</td></tr>
<tr><td> 
</td><td>There are no dependencies between these two resources. The Phantom resource in parallel service groups that do not include on/off resources.</td></tr>
</table>

<table>
<tr><th colspan="2">ClusterService</th></tr>
<tr><td> 
</td><td>The ClusterService service group is built-in. It is created when GCO is enabled. GCO contains a service called the wide area communication (wac) service used for inter-cluster communication. Wac uses a virtual IP on each cluster. This IP must be enabled on each cluster as an AzureIP object since it must be transferred to failover nodes during an interruption in service.</td></tr>
</table>



This is the dependency relationship within the ClusterService service group. Vbsapp is not configured since there is no virtual business service in this solution.

Global Service Group

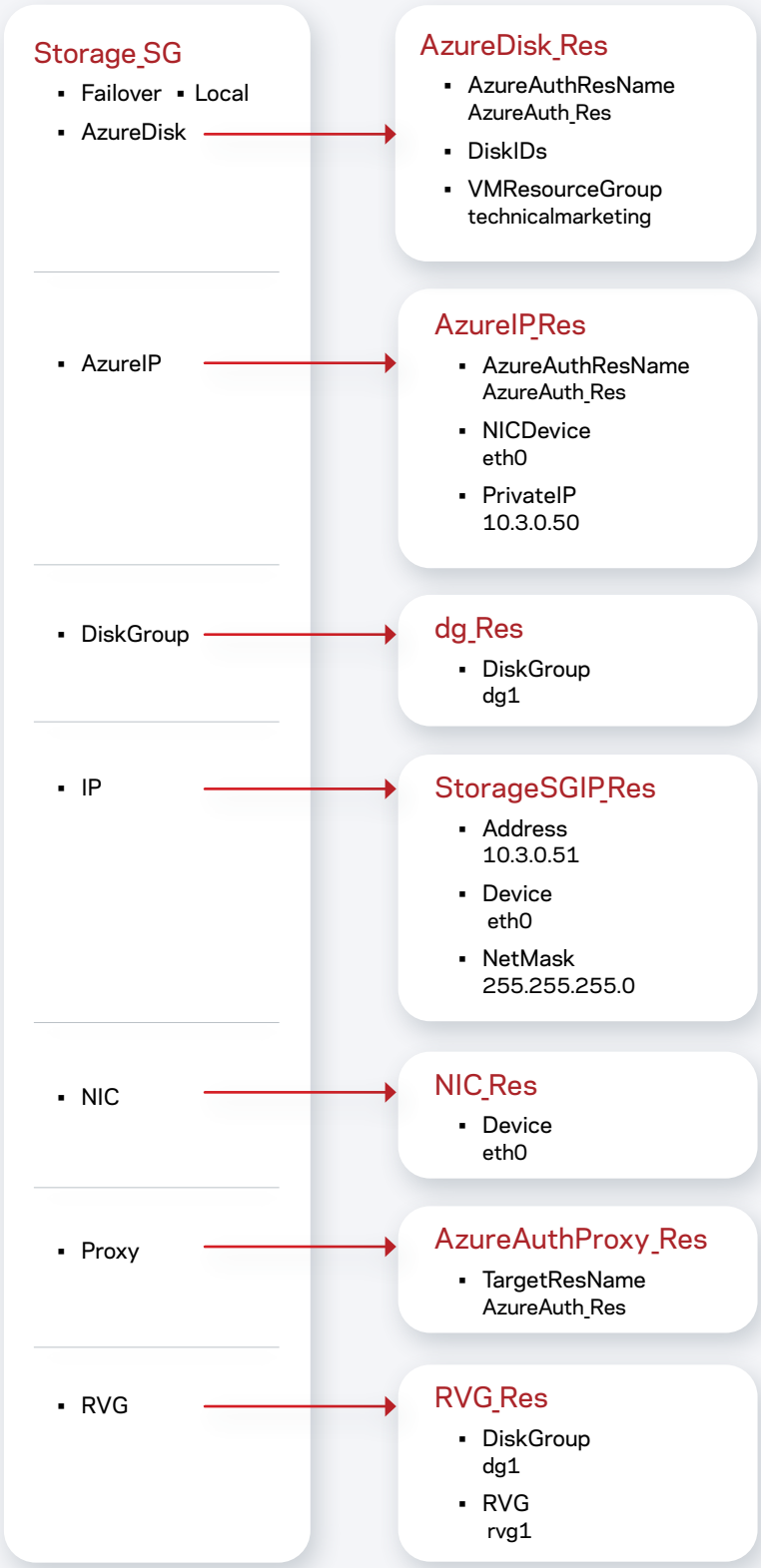


The Global Service Group orchestrates and automates the failover between the separate clusters. It is integral in ensuring that the VVR relationship is preserved and reversed on failback.

Applications requiring Global high availability can be included in this service group.

The mount_Res resource depends on the RVGPri_Res.

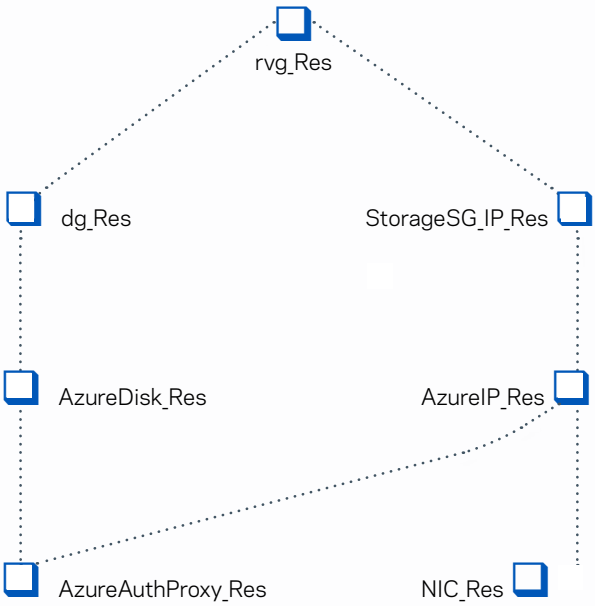
Storage Service Group



The Storage Service Group is not exclusive to managing storage resources. If required, applications that depend on the presence of storage resources must live in this service group.

In general, an application will require the presence of network (AzureIP) and storage (AzureDisk) resources in a single resource group.

Since replication is integral to this solution, the Replication Volume Group (RVG_Res) resource is also included. The RVG_Res resource should be configured after VVR has been set up.



This is how the assorted resource dependencies were configured.

Summary

Veritas is a leader in data resiliency solutions that provide both application and system-level resiliency as well as high availability and disaster recovery for software applications. Veritas Alta Application Resiliency is a proven technology in the marketplace and has evolved to become a premier solution for managing both storage and high availability in public cloud environments. In Microsoft Azure, Veritas Alta Application Resiliency can provide several benefits not available natively that provide the functionality and confidence needed by enterprises looking at Azure as a target environment for applications with the most demanding performance and uptime requirements.

Appendix

Main.cf

```
include "OracleASMTypes.cf"
include "types.cf"
include "CFSTypes.cf"
include "CRSResource.cf"
include "CSSD.cf"
include "CVMTypes.cf"
include "Db2udbTypes.cf"
include "MultiPrivNIC.cf"
include "OracleTypes.cf"
include "PrivNIC.cf"
include "SybaseTypes.cf"

cluster myee67 (
        ClusterAddress = "10.3.0.50"
        SecureClus = 1

        DefaultGuestAccess = 1
        HacliUserLevel = COMMANDROOT
```

```
        )

remotecluster myee89 (
        ClusterAddress = "10.4.0.50"
        )

heartbeat Icmp (
        ClusterList = { myee89 }
        Arguments @myee89 = { "10.4.0.50" }
        )

system myee6 (
        )

system myee7 (
        )

group AzureAuth_SG (
        SystemList = { myee7 = 0, myee6 = 1 }
        Parallel = 1
        Authority = 1
        )

        AzureAuth AzureAuth_Res (
                SubscriptionId = xxx
                ClientId = xxx
                SecretKey = xxx
                TenantId = xxx
                )

        Phantom Phantom_Res (
                )

        // resource dependency tree
        //
        //      group AzureAuth_SG
        //      {
        //      AzureAuth AzureAuth_Res
        //      Phantom Phantom_Res
        //      }

group ClusterService (
        SystemList = { myee6 = 0, myee7 = 1 }
        AutoStartList = { myee6, myee7 }
        OnlineRetryLimit = 3
        OnlineRetryInterval = 120
        )

        Application vbsapp (
                StartProgram = "/opt/VRTSvbs/bin/vbsd"
                StopProgram = "/opt/VRTSvbs/bin/vbsd -stop"
                MonitorProcesses = { "/opt/VRTSvbs/bin/vbsd.bin" }
                RestartLimit = 1
                )

        Application wac (
                StartProgram = "/opt/VRTSvcs/bin/wacstart"
                StopProgram = "/opt/VRTSvcs/bin/wacstop"
                MonitorProcesses = { "/opt/VRTSvcs/bin/wac" }
```

```
        RestartLimit = 3
        )

    AzureIP AzureIP (
        PrivateIP = "10.3.0.50"
        NICDevice = eth0
        AzureAuthResName = AzureAuth_Res
        )

    IP gcoip (
        Device = eth0
        Address = "10.3.0.50"
        NetMask = "255.255.255.0"
        )

    NIC gconic (
        Device = eth0
        )

    AzureIP requires gconic
    gcoip requires AzureIP
    wac requires gcoip

    // resource dependency tree
    //
    // group ClusterService

    // {
    // Application vbsapp
    // Application wac
    //     {
    //     IP gcoip
    //           {
    //           AzureIP AzureIP
    //            {
    //            NIC gconic
    //            }
    //              }
    //     }
    // }

group Global_SG (
    SystemList = { myee7 = 0, myee6 = 1 }
    ClusterList = { myee67 = 0, myee89 = 1 }
    Authority = 1
    )

    Mount mount_Res (
        MountPoint = "/mnt/vol1"
        BlockDevice = "/dev/vx/dsk/dg1/vol1"
        FSType = vxfs
        FsckOpt = "-y"
        )

    RVGPrimary RVGPri_Res (
        RvgResourceName = rvg_Res
        )
```

```
        requires group Storage_SG online local hard
        mount_Res requires RVGPri_Res

        // resource dependency tree
        //
        // group Global_SG
        // {
        // Mount mount_Res
        //      {
        //      RVGPrimary RVGPri_Res
        //      }
        // }

group Storage_SG (
        SystemList = { myee7 = 0, myee6 = 1 }
        )

        AzureDisk AzureDisk_Res (
                DiskIds = {

"/subscriptions/xxxx/resourceGroups/technicalmarketing/providers/Microsoft.Compute/disks/myee67_disk1",

"/subscriptions/xxxx/resourceGroups/technicalmarketing/providers/Microsoft.Compute/disks/myee67_disk2" }
                VMResourceGroup = technicalmarketing
                AzureAuthResName = AzureAuth_Res
                )

        AzureIP AzureIP_Res (
        PrivateIP = "10.3.0.51"
        NICDevice = eth0
        AzureAuthResName = AzureAuth_Res
        )

        DiskGroup dg_Res (
                DiskGroup = dg1
                )

        IP StorageSG_IP_Res (
                Device = eth0
                Address = "10.3.0.51"
                NetMask = "255.255.255.0"
                )

        NIC NIC_Res (
                Device = eth0
                )

        Proxy AzureAuthProxy_Res (
                TargetResName = AzureAuth_Res
                )

        RVG rvg_Res (
                RVG = rvg1
                DiskGroup = dg1
                )

        AzureDisk_Res requires AzureAuthProxy_Res
        AzureIP_Res requires AzureAuthProxy_Res
```

```
        AzureIP_Res requires NIC_Res
        StorageSG_IP_Res requires AzureIP_Res
        dg_Res requires AzureDisk_Res
        rvg_Res requires StorageSG_IP_Res
        rvg_Res requires dg_Res

        // resource dependency tree
        //
        // group Storage_SG
        // {
        // RVG rvg_Res
        //     {
        //     IP StorageSG_IP_Res
        //             {
        //     AzureIP AzureIP_Res
        //             {
        //             Proxy AzureAuthProxy_Res
        //             NIC NIC_Res
        //             }
        //     }
        // DiskGroup dg_Res
        //     {
        //     AzureDisk AzureDisk_Res
        //             {
        //             Proxy AzureAuthProxy_Res

        //             }
        //        }
        //      }
        // }

group cvm (
        SystemList = { myee6 = 0, myee7 = 1 }
        AutoFailOver = 0
        Parallel = 1
        AutoStartList = { myee6, myee7 }
        )

        CFSfsckd vxfsckd (
                ActivationMode @myee6 = { myee67_fss_dg1 = sw }
                ActivationMode @myee7 = { myee67_fss_dg1 = sw }
                )

        CVMCluster cvm_clus (
                CVMClustName = myee67
                CVMNodeId = { myee6 = 0, myee7 = 1 }
                CVMTransport = gab
                CVMTimeout = 200
                )

        CVMVxconfigd cvm_vxconfigd (
                Critical = 0
                CVMVxconfigdArgs = { syslog }
                )

        ProcessOnOnly vxattachd (
                Critical = 0
                PathName = "/bin/sh"
```

```
        Arguments = "- /usr/lib/vxvm/bin/vxattachd root"
        RestartLimit = 3
        )

    cvm_clus requires cvm_vxconfigd
    vxfsckd requires cvm_clus

    // resource dependency tree
    //
    // group cvm
    // {
    // ProcessOnOnly vxattachd
    // CFSfsckd vxfsckd
    //     {
    //     CVMCluster cvm_clus
    //         {
    //         CVMVxconfigd cvm_vxconfigd
    //          }

    //     }
    // }
```

o

References

Farhat, S. Azure Managed vs Unmanaged disks: The Choice:
https://buildwindows.wordpress.com/2017/05/31/azure-managed-vs-unmanaged-disks-the-choice/

Microsoft Azure Virtual Network Documentation:
https://docs.microsoft.com/en-us/azure/virtual-network/

Introduction to Azure Managed Disks:
https://docs.microsoft.com/en-us/azure/virtual-machines/windows/managed-disks-overview

Veritas Cluster Server 8.0 Administrator's Guide:
https://sort.veritas.com/DocPortal/pdf/79561893-149457104-1

Veritas Cluster Server 8.0 Bundled Agents Reference Guide - Linux:
https://sort.veritas.com/DocPortal/pdf/79620650-149457740-1

Veritas InfoScale 8.0 Replication Administrator's Guide - Linux:
https://sort.veritas.com/DocPortal/pdf/79604030-149462309-1

Veritas Technologies. (n.d.). Veritas InfoScale 8.0 Solutions in Cloud Environments:
https://sort.veritas.com/DocPortal/pdf/130803809-149463390-1

Storage Foundation Cluster File System High Availability 8.0 Configuration and Upgrade Guide - Linux:
https://sort.veritas.com/DocPortal/pdf/79735435-149461812-1

Veritas InfoScale 8.0 Disaster Recovery Implementation Guide - Linux:
https://sort.veritas.com/DocPortal/pdf/79901122-149461867-1

About Veritas

Veritas Technologies is a leader in multi-cloud data management. Over 80,000 customers—including 95 percent of the Fortune 100—rely on Veritas to help ensure the protection, recoverability, and compliance of their data. Veritas has a reputation for reliability at scale, which delivers the resilience its customers need against the disruptions threatened by cyberattacks, like ransomware. No other vendor is able to match the ability of Veritas to execute, with support for 800+ data sources, 100+ operating systems, 1,400+ storage targets, and 60+ clouds through a single, unified approach. Powered by Cloud Scale Technology, Veritas is delivering today on its strategy for Autonomous Data Management that reduces operational overhead while delivering greater value. Learn more at www.veritas.com. Follow us on Twitter at @veritastechllc.

VERITAS

2625 Augustine Drive
Santa Clara, CA 95054
+1 (866) 837 4827
veritas.com

For global contact information visit:
veritas.com/company/contact

V0996 03/23